A New Direction in Data Authentication - Single Cycle Massive Diffusing DRNGs

Carmi Gressel carmi@fortressGB.com
Orr Dunkelman orrd@cs.technion.ac.il
Avi Hecht avi@fortressGB.com
Ran Granot ran@fortressgb.com

Abstract: Good Random Number Generators, RNGs, Stream Ciphers, SCEs, and Hash functions have much in common, and many cryptographers claim that they are all equivalent random functions. The recently discovered weaknesses found in popular Hash and MAC, mechanisms, is triggering renewed use of popular symmetric ciphers for hashing documents. We have conceptually done the obvious, using disparate pseudo-random devices working in tandem to give top quality random numbers on every machine cycle, integrated into the 3 modes. A procedure that is unpredictable, with massive diffusion of single data bits into the "guts" of a well constructed pseudo-random binary generator with massive diffusion, that is "easy enough" to analyze in order to realize the intractability of efforts to efficiently compromise the procedure, is the natural candidate for next generation Data Authentication encoding. Being that the ZK-Crypt is an extremely robust, fast, energy conservative implementation of a one way function, it is suited for all phases of symmetric protection of data. The standard device is small enough to be put into a smart card, strong and fast enough to go into a main frame; and with 334 binary variables has high crypto-complexity and can supports large keys. [1][4][5]

Preface- why a New Direction:

Now is the time to implement the best strong solutions for the three random number generating functions – possibly the most important, a large entropy store guaranteeing collision resistant data authentication; a true random number generator, with online proof of the validity of the noise source; and a strong stream ciphering device with long running keys and no error propagation. We demonstrate our low cost, low energy consuming device for up to 7 Giga bits per second and more throughput.

The ZK-Crypt is "a long time in the making" high entropy multi-purpose holistic random number generator. The architecture, the disparate randomizing methods, and the concept have been extensively reviewed, rigorously tested and analyzed.

The ZK-Crypt eSTREAM version was implemented on silicon, at the Integrated Systems Laboratory at the ETH along with seven other contestants [8]. Only three contestants had high throughput, and reasonable size. The ZK-Crypt was the only one of the three that had long keys (the other two had 80 bit secret running keys). Only the ZK-Crypt was designed for efficient Hashing.

David Naccache writes to us after his review:
"The hardware characteristics of the stream cipher/data authenticator/true random number generator fit the requirements of future generation smart cards: low power, reduced critical paths, only 8K gates - where all this boils down to is a secure 3 Gb/s throughput when clocked at 100 MHz. Being AIS 31 compliant paves the way to a valuable cheap True Random Number Generator."

Fortress U&T and FortressGB's track record with a large family of best selling crypto-processors is based on secured hardware access to crypto-functions; with no external access to keys; "if you want to keep a secret- don't know it". We believe that efficient strongest product is best achieved in hardware. FortressGB has released a compliant software version for legacy applications.

Introduction – an 8K Gate Versatile Engine:

As hackers needle into every nook and cranny of our boots, our automobile control systems and our confidential files; and as the smallest portable devices communicate with main frames, and satellites; a fast very low power secured hardware data authentication module is a first line of defense.

The ZK-Crypt Data Authentication module is based on an extremely small, high throughput single cycle 32 bit random number generator. The Deterministic Random Number Generator (DRNG) generates top mark DieHard strings and is tailored to the needs of collision proof data authentication. With less than 8000 logic gates the ZK-Crypt attains a degree of non-linear complexity difficult to achieve efficiently with standard microprocessors; e.g., the ZK-Crypt processes 3 Gb/second at 100 MHz, and has been tested at 250 MHz.

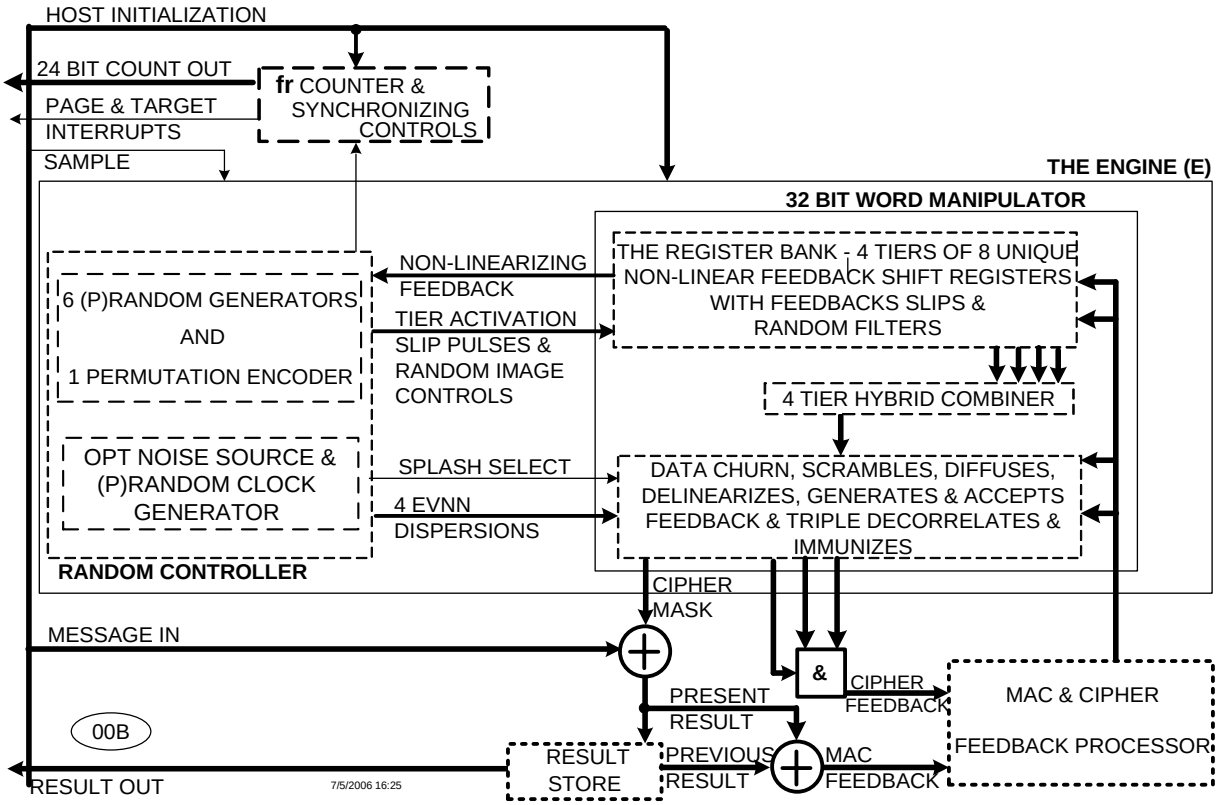

Figure 1- An overview of the ZK-Crypt engine and Ancillaries [1]

The Data Authentication module operates equally well as an unkeyed Hash or as a loadable keyed Hash (aka MAC) device. Keys and tags may be any arbitrary length; e.g., from 128 to 512 bit. With its random Frequency Modulated noise source the module is an AIS-31 True Random Number Generator [3]. In deterministic cipher mode, the ZK-Crypt is a high throughput page synchronized long key stream cipher.

Interestingly, the European eSTREAM stream cipher contest is looking for a combination hardware Stream Cipher and Data Authentication Encoder (MAC), whereas NIST has "flung down the gauntlet" challenging the security engineering community to devise a combination Data Authenticator and True Random Number Generator (TRNG). The ZK-Crypt executes all three functions using the same tested and analyzed DRNG [4][5].

An Overview of the Processor:

The ZK-Crypt engine is divided into two interactive main modules and two ancillaries as shown in Figures 1 and 4 -

1) The 32 bit Word Manipulator consisting of a Register Bank and a Data Churn. The Register Bank consists of 8 unique non-linear feedback shift registers, organized in four tiers, aberrated by feedback slips, and with rotated randomly XORed images. The 4 32 bit words of the Register Bank are combined in hybrid MAJ/XOR filters. The Data Churn subsequently 3 time stores and triple XORs previous and present permuted outputs, twice filters the 32 bit words through 4 rule 32 bit displacement matrices, wherein each displaced output bit from each matrix is combined in four specific instances into 4 adjacent hybrid filters. The filters are regulated by clocked permutation controllers.

Non-Linear Feedback Shift Registers, nLFSRs, typically, are at the heart of Deterministic Random Number Generators, and Stream Ciphers. The nLFSR configuration in Figure 2 is typical of the TOP, MID and BOT tier nLFSRs in the Register Bank.

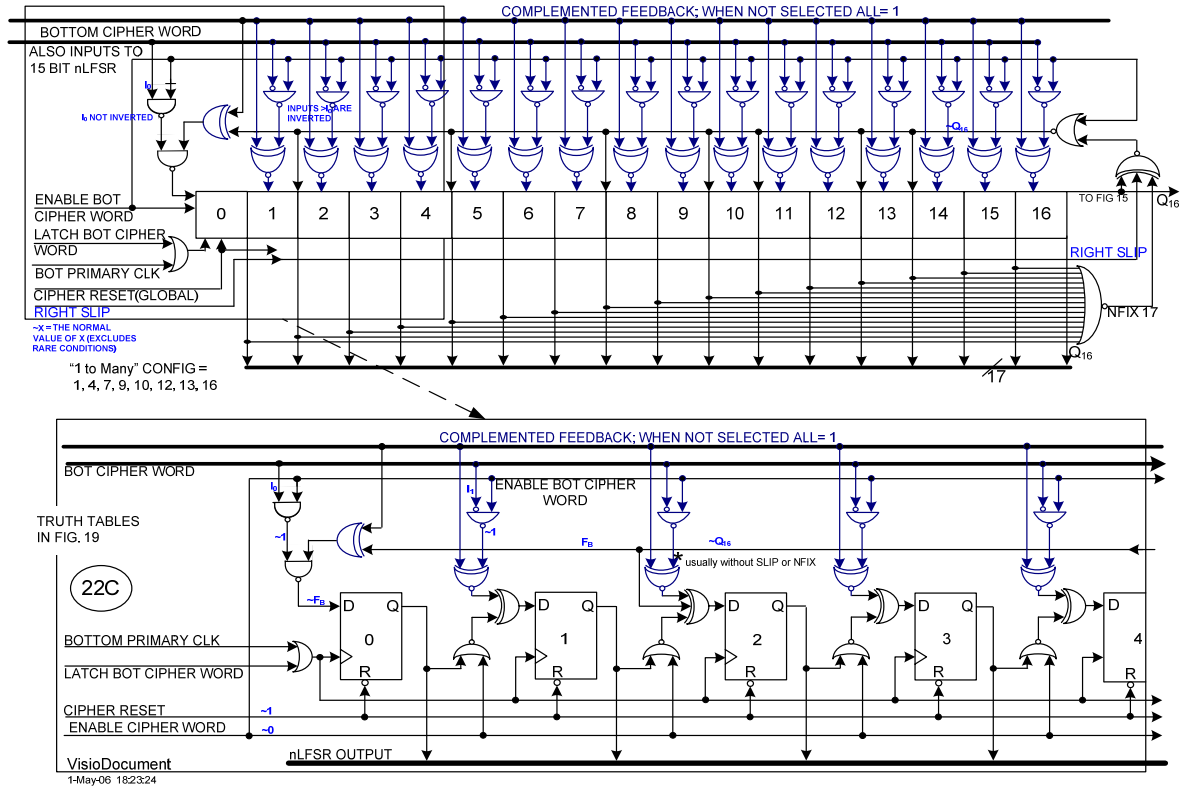

Figure 2 – One of 8 non-Linear Feedback Shift Registers, with pre-Load, Slip, and Parallel Feedback [1]

The heart of the Register Bank consists of 8 unique One to Many (Galois) maximum length non-Linear Feedback Shift Registers, nLFSRs; or simply Feedback Shift Registers, FSRs. All 8 FSRs are operative to receive parallel feedback; the Top, Middle and Bottom tier FSRs can be explicitly initialized (Loaded), and also are operative to receive random Slip aberrations on the serial feedback, see Figure 2. All variables in a deterministic mode are Reset initially, prior to initializing with running keys. Note the NFIX 17 function, operative to output a "1", iff flip-flops 0 to 15 (n-1 LS cells) are at logic "0". This precludes the danger: a) of a "stuck on zero" nLFSR, as an ordinary LFSR ceases to operate when all "n" cells outputs remain at logic zero and b) alternately inserts the all "zero stage" to balance occurrences of "1"s and "0"s.

If an adversary learns 2m consecutive bits from a normal m celled LFSR output, she/he can easily predict future sequences. The Slip and Feedback aberrations preclude a sequence of 2m consecutive bits from the normal nLFSR sequence, thereby foiling formal cryptanalytic attacks.

In Figure 3, the Bottom Tier of the Register Bank of Figures 4, 5 and 6, the two nLFSRs are activated by the Random Controller, and are operative to receive feedback and random Slip pulses. The Tier also generates a 5 Left Rotated image. The image is randomly XORed to the nLFSRs' parallel outputs. If an adversary could see the output of the two nLFSRs, she/he could sense a left to right movement. The XORed output of the nLFSRs and the image serves to partially mitigate the sense of left to right oriented motion. As the parallel feedback and the Slip pulses typically bias the nLFSR output, the XORed image vector also serves to reduce the biases to "1" or "0" by an order of magnitude.

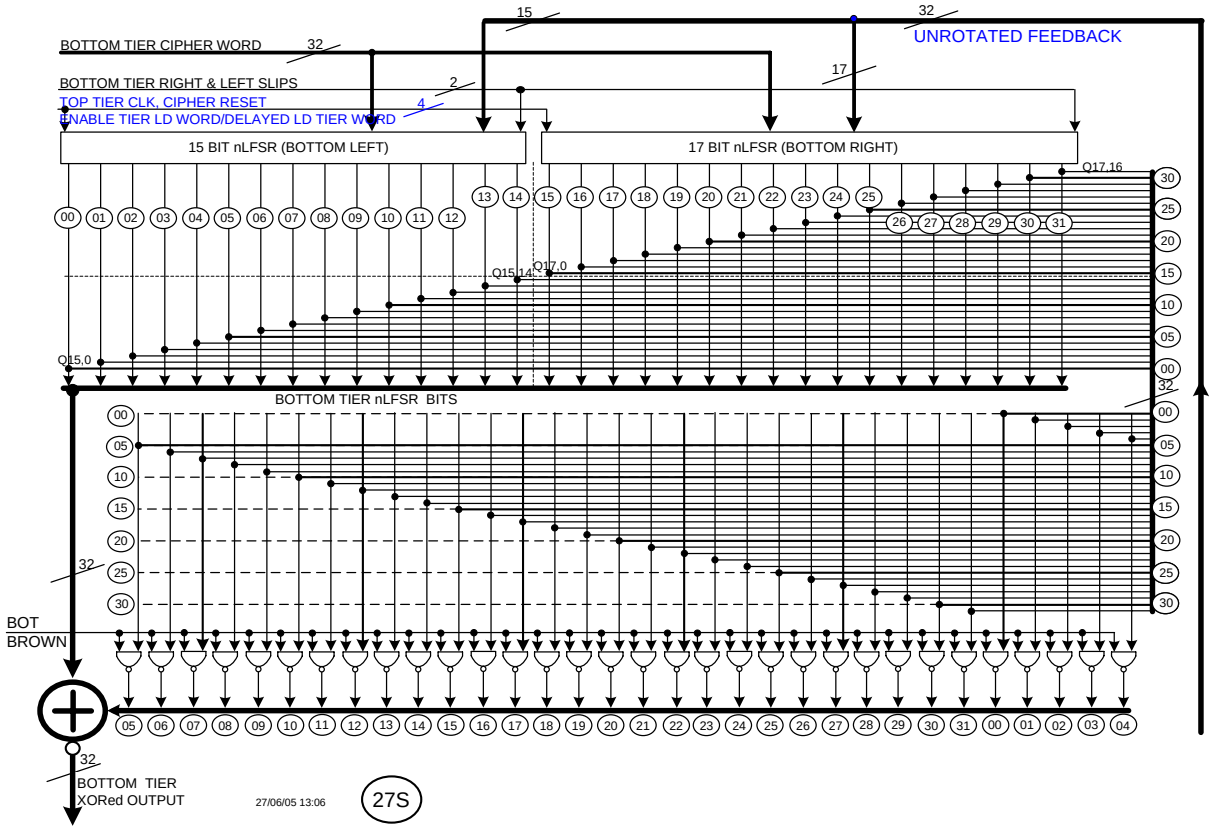

Figure 3 – The Bottom Tier with 15 and 17 bit nLFSRs, with Rotated Image, randomly XORed to nLFSR output [1]

Figure 3 depicts a full tier "assembly". Upper left and right blocks are the two nLFSRs, the bottom half is the 5 left rotated image. When the Bot Brown signal is a "1" the image is XORed to the outputs of the nLFSRs. The TOP, MID and BOT tiers are configured similarly.

The Super Tier's rotated image is always XORed to its nLFSRs outputs (and is not explicitly loaded by the Host). Therefore, the Super Tier's output is consistently less biased, and serves to improve the output statistics of the MAJ/XOR Hybrid Combiner output in Figure 4 as the 32 bit word "ripples down" improving statistics at every junction.

In Figure 4 we see how the three tiers are combined in a Hybrid MAJ/XOR 32 bit filter. Here a 2 of 3 Majority (MAJ) non-linear combining vector accepts the outputs of the TOP, MID and BOT tiers. The MAJ output image is a 5 Right Rotated vector which is XORed to the output of the MAJ vector; and the result is again XORed to the output of the Super Tier.

We can assume that the output of the MAJ combiner will be biased, as a result of the Slip and the Feedback skewing, but that the output of the Super Tier will be less aberrated. The Triple XOR leading to the Top Store & XOR will largely improve the statistics. Unbiased statistics are most important for Stream Ciphering and Data Authenticating Tagging, see Figure 9.

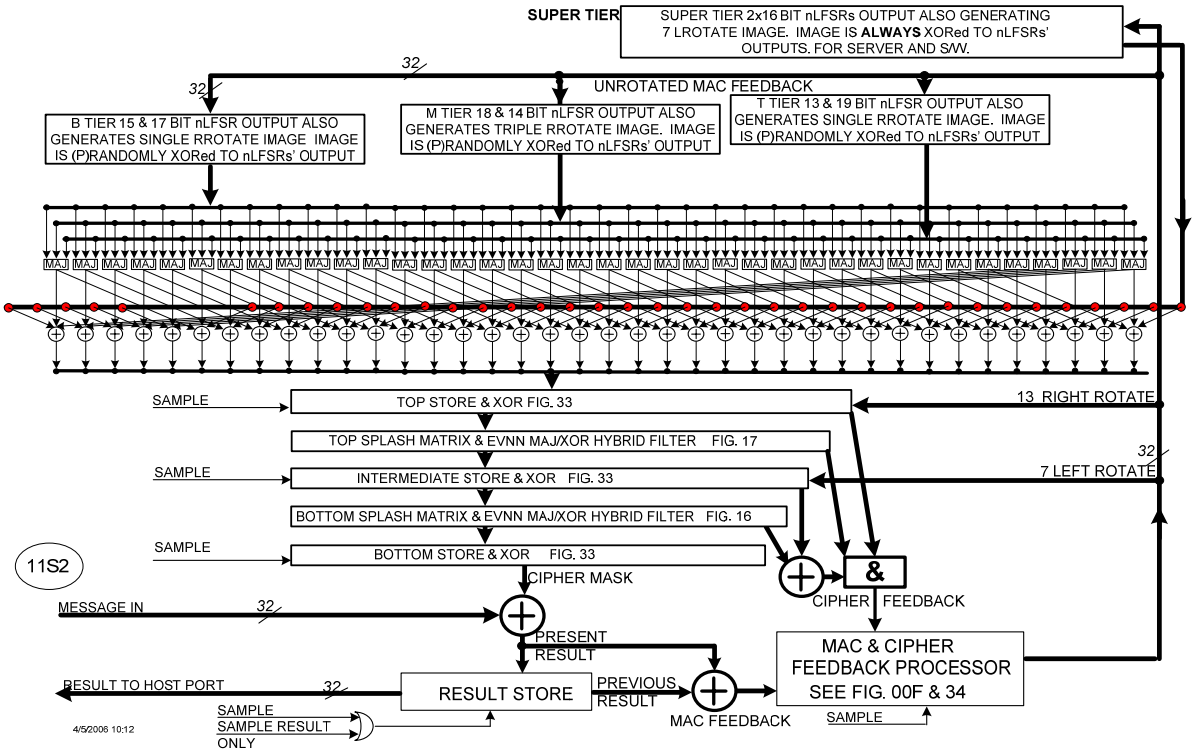

Figure 4 – MAJ Combining the Top, Mid and Bot Tiers, XORed to the 5R Rotate & the Super Tier output [1]

2) The Random Controller consists of two main modules. The Control Units (each of the 3 units has two unique pseudo random generators), permutation encoders and the combination deterministic pseudo-random clock/ noise source clocking device. The Random Controller at every step reconfigures the permutations in the Word Manipulator, and accepts non-linearizing feedback from the Register Bank variables. Said differently, changing data in the Register Bank randomizes the Random Controller.

Figure 5 pictures the interaction between the Random Controller and the Data Manipulator. There are over 15 permutation controlling signals emanating from the Random Controller to the Data Manipulator, and only five feedback signals disbursed from the Manipulator back to the Random Controller. This means that data changes are reflected into the Random Controller with latency whereas the control signals cause immediate complement and displacement changes in the Word Manipulator. The Cipher and MAC feedback at each machine cycle "ripple" down and affect the Register Bank, the Data Churn, the Result Store and Feedback Store on the next clocked cycle. The Top, Intermediate and Bottom Stores & XORs are likewise affected by the feedback and these changes rain down to the Result and Feedback Store. In many respects, this recycling and rotating is similar to the pseudo-entropy developed in modular squaring of a random number over a prime.

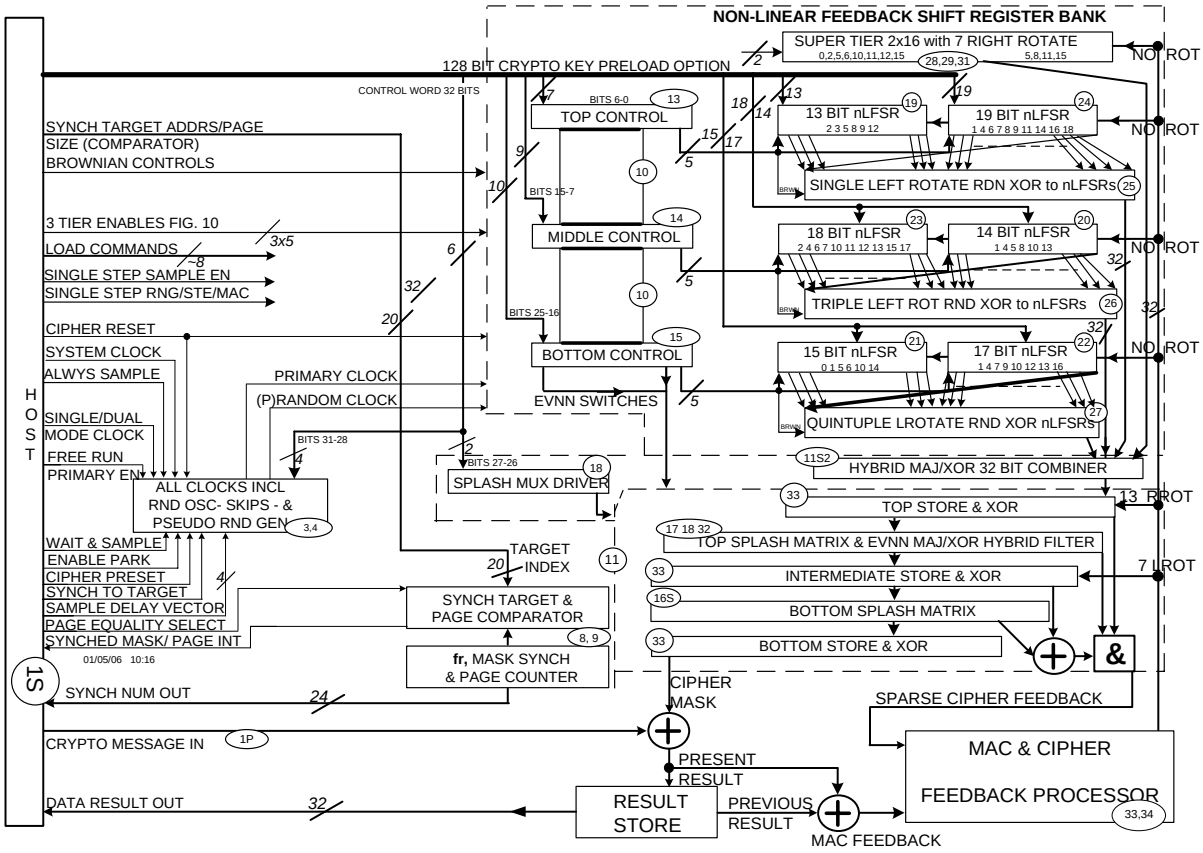

Figure 5. The Actors in place (encircled numbers refer to drawings in [1])

The ancillary units:

3) The 32 bit MAC and Cipher Feedback Processor accepts message words, and processes feedback as prescribed for Ciphering and Tagging and for keyed and unkeyed Hashing. See Figure 9.

4) The **fr** Counter and the Synchronizing Controller automate page synchronization for long file transmissions. In the True Random Number Generation mode the Host monitors wandering phase differentials between the random frequency modulated noise source and the Host supplied Primary Clock. [3]

Feedback Strategy

Those practiced in the art know that even single bit random aberrations to maximum length Linear Feedback Shift Register (LFSR) sequences degrade output statistics. No less, parallel dense feedback of intermediate results into LFSR stages and other permutations, if not properly compensated, can devastate the output statistics of any random sequence generator. Intelligent generation of feedback for ciphering and random number generating can assure full diffusion of sparse aberrations with minimal or even no statistical degradation, assuming that adequate correlation immunity and bias removal compensations are enacted. See Figure 7.

Conversely, in a well designed deterministic engine, non-observable, internally generated feedback helps to make the cryptanalyst and hacker's tasks virtually intractable. In the ZK-Crypt, a small average number of unpredictable "1" bits in the feedback will each diffuse into every ZK-Crypt variable with minimal distinguishable statistical degradation on the output, because of the extensive and judicious use of Hybrid MAJ/3XOR and Store & XOR filters as in Figures 4,7 and 8.

Contrary to the constraints limiting the amount and observability of feedback in ciphering, the target of data authentication digesting feedback is to maximally diffuse every bit of the data file into each of the 334 binary

variables of the of the authentication processor. Over 300 binary variables store "entropy" to support any practical length Data Authentication Tag.

Three different rotated and un-rotated feedback images are "recycled" into the data manipulator. At least three of the tiers accept the un-rotated vector image of the feedback, which causes an "unforgettable" change of stages in the activated feedback shift registers. 7 and 13 bit rotated vectors cause transient single clock aberrations in the data churn (which are recycled in the following machine cycle via the feedback).

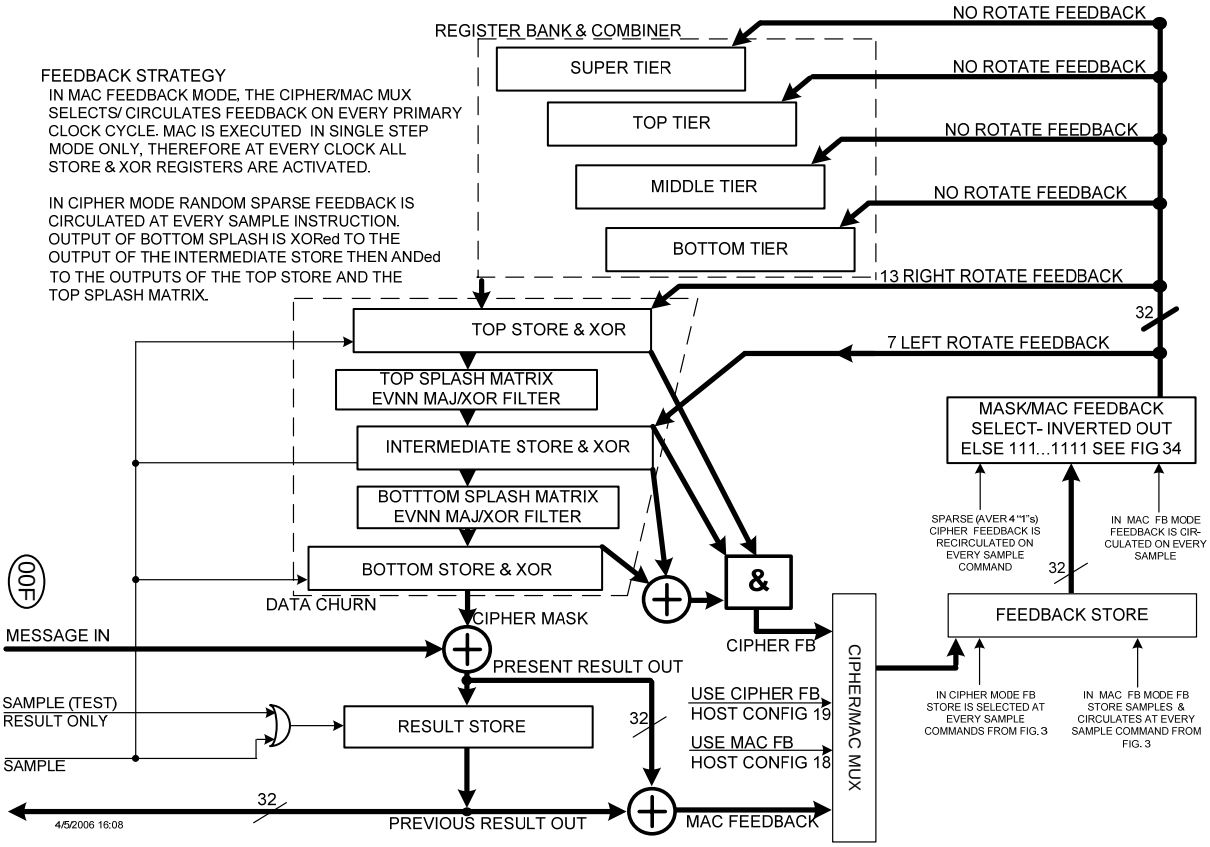

Figure 6 - The Feedback Strategy - Maximum for Data Authentication Digesting and Initialization [1]

Sparse cipher feedback as used in the Hash Initialization, Interhash Cipher Feedback Digest and the subsequent tag generation (also in Stream Ciphering and TRNG processing) is drawn from four consecutively increasingly crypto-complex sources (climbing the ladder of Data Churn). The output of the Bottom Hybrid Filter and the Intermediate Store & XOR decorrelator are XORed together. This output and the outputs of the Top Hybrid Filter and the Top Store & XOR are combined in a 3 input AND vector to form the Sparse feedback vector. The result is that an average of four random "1"s will appear in the feedback vector, and only rarely will no "1" appear in the feedback vector. Such an occasional null vector is not problematic, as previous feedbacks have "long lasting" unpredictable traces. Later, we show that even a single "1" aberrates almost the entire cipher mask in the next Sample command, and after one more Sample command virtually each variable in the cipher mask (the output of the Bottom Store & XOR buffer) is affected, as are most of the other 128 variables in the Word Manipulator. Data changes serially ripple from the Word Manipulator to the Random Controller via 5 serial signals on the interface.

Compensating the Statistic Feedback "Hump"

Trade-off precautions and defense mechanisms are necessary to decorrelate (correlation immunize) and to remove bias generated by aberrations which bias otherwise pseudo random sequences. These compensating mechanisms include: "Store & XOR" of biased and slightly uncorrelated binary streams, wherein a memory device (a flip-flop) outputs the previous input which is XORed to a present input bit; hybrid filters wherein five possibly loosely correlated variable bits are combined by non-linear two of three majority gates (MAJs) and 3 exclusive OR gates; XOR vector combiners joining random streams which rotate in one direction; e.g., Register Bank feedback shift register (FSR) outputs are rotating to the right, while the image streams are rotated in a second direction, e.g. left rotate, to reduce, reverse or eliminate the natural direction orientation of the three tiers of FSRs.

For data digestion, the above compensating mechanisms, when implemented repeatedly and judiciously, (the equivalent of many rounds in a Feistal machine), evoke accelerated diffusion of each bit of an input "message" word into all variables of the Register Bank and the Data Churn.

If we run the ZK-Crypt as a DRNG without feedback, and test the output with a full scale DieHard suite [7], we get top marks. [Usually only one p value close to either zero or one, e.g., 0.00nϕϕ or 0.99mϕϕ where n>0 and m<9.] With feedback, we rarely receive top marks but never failed on any one of the over 200 DieHard test functions. From this we presently infer that the slightly degraded statistics distinguish between DRNGs with and without feedback, only. This inference is strengthened by our previous work on DRNGs based on B^2 mod N and B^2 2^{-512} mod N (Montgomery multiplication), where B and N were colored random strings with $2^{512}>N>2^{511}$. Hamming weight statistics of B^2 mod N were slightly degraded when B was raised to the 3rd power.

Hybrid Filters with Store & XOR Decorrelation

An equivalent implementation of a 2 of 3 majority logic gate is shown in the upper left corner of Figure 7. Figure 7 depicts the diffusion (seen also in i'th filter cell of Figure 8) and the decorrelation affected by the flip-flop of the Intermediate Store & XOR. A similar structure appears under the Bottom Splash Matrix.

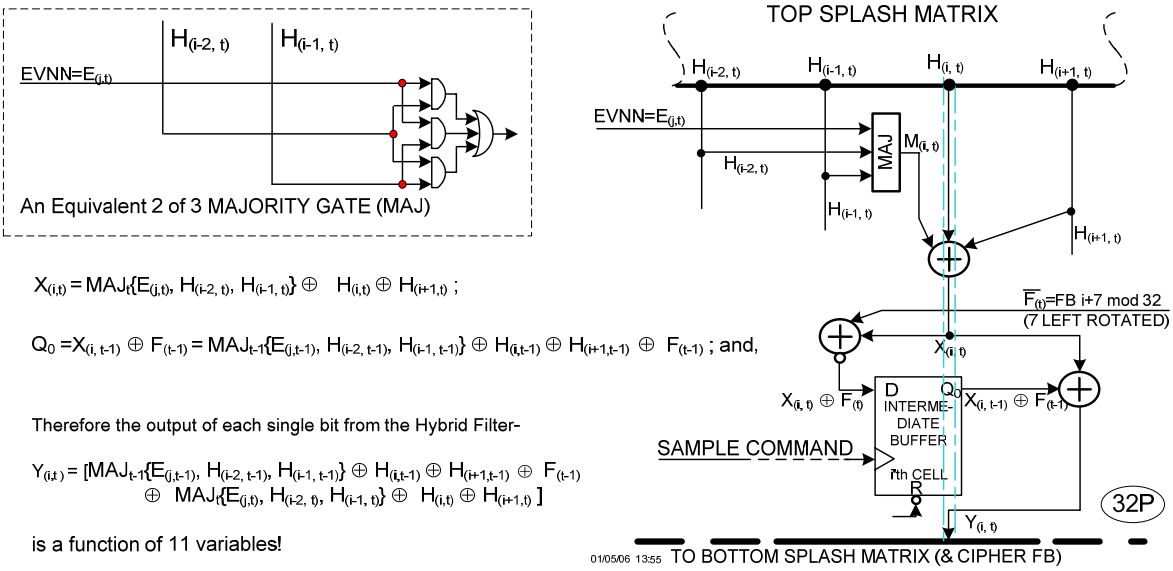

Figure 7 - An 11 Bit Variable Diffusion in the Hybrid MAJ/3 XOR Filter [1]

The 2 of 3 Majority Gate, MAJ, is a 3 input non-linear combining function useful for decorrelating input variables. Non-linear functions, e.g., MAJ, AND and Carry, encumber the cryptanalyst's resources, as matrix equations become far more complicated with non-linear functions, than when the cryptanalyst deals only with linear XOR gates and LFSR type pseudo-noise generators.

If the three MAJ input bits are unbiased, then the output statistics are unbiased with output probability of one half. However, in small intervals (and occasionally in large intervals) variables are often biased. The problem with MAJ functions is that only in rare instances do the MAJ gates significantly lower the cumulative bias caused by the input bits. The 3XOR function, on the other hand, typically lowers the output bias of uncorrelated input signals to less than the bias of the least biased input bit. In the extreme, remember that a constant "1" XORed to an unbiased stream generates a complemented unbiased stream; either a constant "1" (or constant "0") input would devastate a MAJ processed sequence; a "0" output on the average of only 1/4.

By combining the two fairly unbiased inputs, $H_{(i,t)} \oplus H_{(i-1,t)}$, in Figure 7, which we have good reason to believe are uncorrelated because of previous Splash displacements and permutations (see the Splash Matrix in Figure 8), we can assume that $X_{(i,t)}$ is significantly less biased than either $H_{(i,t)}$ or $H_{(i+1,t)}$.

The EVNN $E_{(j,t)}$ signal, $0 \le j \le 3$, is one of 4 typically unbiased permutation configured control signals. The inverted feedback signal, F(t) is definitely assumed to be extremely biased. Typically F(t) alone does not adversely affect the input to the flip-flop.

The equation from Figure 7 models the output of each single bit input of the Hybrid Filter –

$$Y_{(i,t)}=[MAJ_{t-1}\{E_{(t-1)}, H_{(i-2,\,t-1)}, H_{(i-1,\,t-1)}\} \oplus H_{(i,t-1)} \oplus H_{(i+1,t-1)} \oplus F_{(t-1)} \oplus MAJ_t\{E_{(t)}, H_{(i-2,\,t)}, H_{(i-1,\,t)}\} \oplus H_{(i,t)} \oplus H_{(i+1,t)}\ ].$$

Eight of the outputs are diffusions from the Splash Matrix, each carrying the memories of previous feedbacks "never forgotten" from the four tiers of the Register Bank, cycled and recycled, imaged and repeated. Two of the "taints" are from the typically unbiased EVNN control bits, which also reflect diffused traces of previous feedbacks.

At each Sample command, the Super Tier is aberrated by each "1" bit in the Feedback word. Likewise the Super Tier 7 Left Rotate image is XORed to the nLFSRs' outputs. At each Sample command, one of the three (TOP, MID or BOT) tiers may not be activated (with a probability of one half). Unactivated tiers do not incorporate feedback words, but will "remember" the cumulative effects of previous activations, and will output its previous output with aberrations. The 1, 3 and 5 Left Rotated images are randomly XORed to the three tiers' nLFSRs' outputs, regardless of if the tier is or is not activated.

A lone feedback i'th bit will "taint" single i'th index flip-flops in three or four tiers, always "taints" the i'th flip-flop of the Super Tier. This bit will always "taint" the i'th input and output of the 3 Tier MAJ Combiner, and will also "taint" the (i+5)'th bit XORed rotated image output of the MAJ combiner. As the Super Tier is always 7 Left Rotated and XORed to its image; in the Hybrid Combiner, the lone "1" i'th bit input to the Super Tier will always also "taint" the (i-7)'th bit of the Hybrid Combiner's output.

Minimally, we see that if only two or three of the TOP, MID or BOT tiers are activated at the sample clock, and only one of their images is XORed, three output bits of the Hybrid Combiner (the i'th-7, the i'th and the i'th+5) bits and another from the single XORed image will be tainted at the next machine cycle by the lone i'th "1" bit of the feedback, as both the Super Tier's nLFSRs and Rotate Vector and the Hybrid Combiner of Figure 4 are always activated.

Maximally, we see that if all four tiers are activated, and their images are XORed to the outputs of their nLFSRs, then the (i'th+5)'th, the (i-1)'th, the (i-5)'th and (i-7)'th bit will be tainted. The probabilities are not all equal for 3, 4, 5, or 6 tainted outputs from a single "1" in the feedback. We have seen that the Hybrid Combiner's output rains down to the Result on the first machine cycle, and also the adjacent bits of each aberration in the Register Bank are typically affected in the following machine cycle, as the bits are right shifted at each tier activating machine cycle. Now we can agree that taints are effective for at least two machine cycles, and the single bit "taint" affects two output bits if there is a rotated image, and there are, on an average of four "taints" in the Super Tier feedback alone. There is an additional maximum of another 12 tainted bits from the TOP, MID and BOT tiers from a lone "1" in the Feedback.

Assuming a "1" on indexed bit 16, a typical sized mix of affected bits from the Hybrid Combiner might be-

TNT= **9**, **10**, **16**, **17**, 19, 20, **23**, **24**; the bits in bold are always affected, as the Super Tier and Hybrid images are always active. Note, a taint may be invisible; the important consequence is that the Register Bank is indelibly tainted.

The same lone index 16'th bit of the Feedback Word also affects the (16+13)'th column of permutations via the Top Store & XOR, because of the 13 Right Rotate, and the (16-7)'th Feedback bit which affects the Middle Store & XOR.

At each Sample, we have a spread of about 7 taints from the Register Bank to the Top Store & XOR. 8 inputs from the Top Store & XOR taint each Hybrid bit Intermediate Store & XOR, as does the lone (16+13)'th shifted feedback bit. We assume that the number of tainted outputs from a lone feedback "1" in the first stage is amplified, again and again in the second phase.

We assume that on the average with Sparse feedback, taking into account the cumulative effect of previous feedbacks, that the Sparse feedback protocol is more than sufficient to mask an output on the Cipher Mask. Note, the chance of an all "0" Sparse feedback word is 1.4%, and the occurrence of two consecutive all "0" Sparse feedback 32 bit words is 2 in 10,000.

The Four Rule Displacement Splash Matrices

The Splash Matrices e.g., the Top Splash Matrix in Figure 8, typically change the mix of the variables which are input into the hybrid filter. For example, the four mixes of Splash inputs to the index 16 hybrid filter are, as per rules A, B, C, & D:

A(06, 03, 17, 13), B(27, 21, 14, 24)
C(26, 31, 08, 06) and D(15, 16, 17, 17).

At each Sample command the random select of the next rule (A, B, C, or D) is irrespective of the previous choice. The Top and Bottom Splash rules are never the same, the EVNN permutations on the two matrices are also never the same and the feedback vectors are not the same because of the different rotations (13 Right and 7 Left) of the output of the Feedback Processor. Therefore, the 11 factor serial output from the Top Splash Matrix via the Intermediate Buffer is diffused again such that another 11 factors are typically added into the output of the Bottom Store & XOR buffer.

Such massive random diffusion in software would entail tens of machine cycles of a 32 bit CPU.

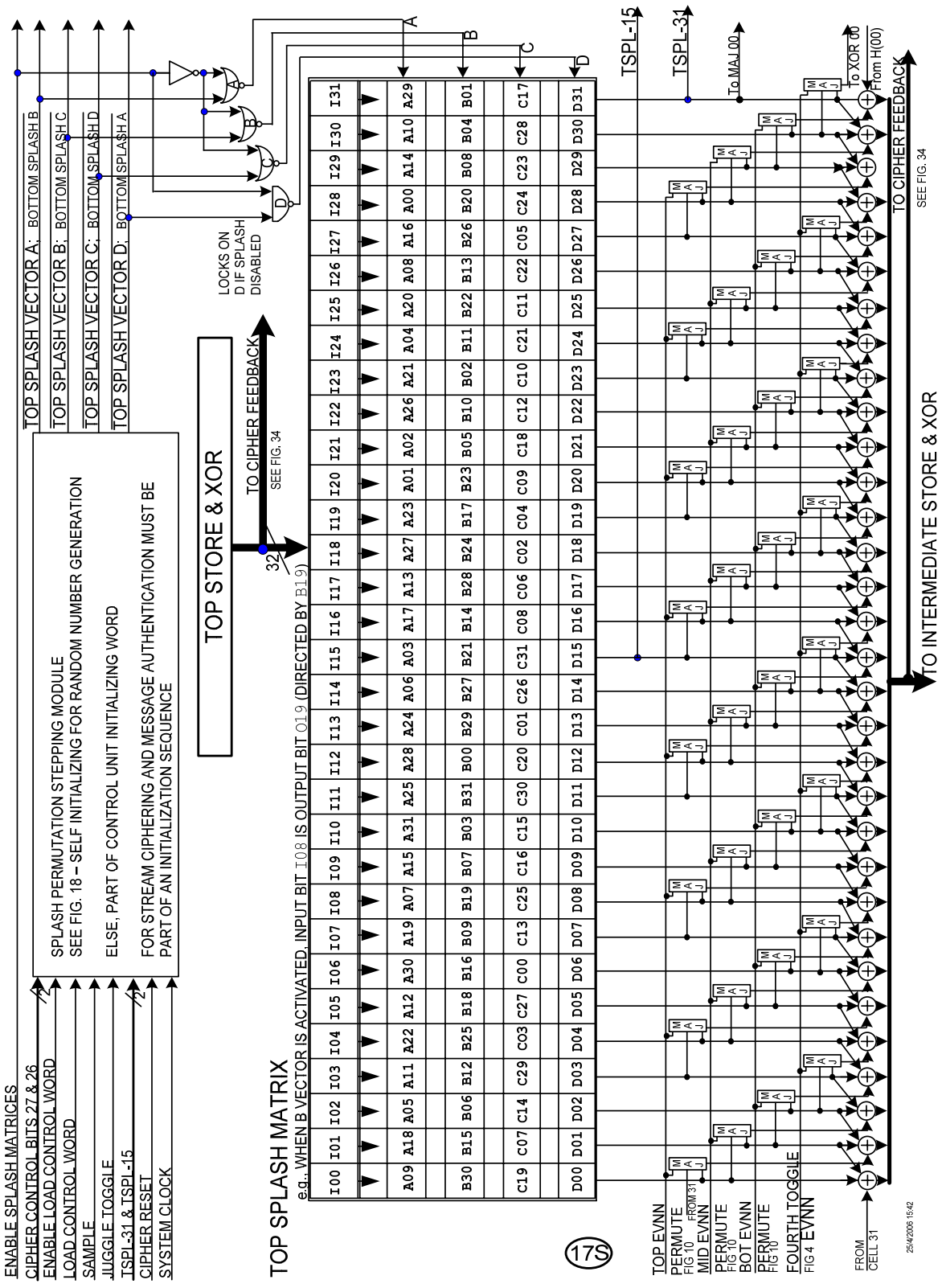

Figure 8- The Top Splash Matrix with the Top Hybrid MAJ/3XOR Filter [1]

Feedback Signal Diffusion via the Register Bank

The immediate effect of a single feedback bit in an activated Tier (at least 2 of the three Top, Middle or Bottom are activated at each Sample) immediately trickles down to the Data Churn into the non-linear MAJ 3 tier combiner.

If the activated tier is rotate/imaged, the feedback bit affects two of the tier's output bits. When the 3 tiers are activated simultaneously (a probability of 1/2) and all images are XORed to the source FSRs, then 4 bits of the MAJ combiner outputs are affected, and are input and "amplified" in the Top Store & XOR, prior to being further amplified by the following double complexes of Splash Matrix and Store & XOR buffer. Remember that traces of the previous MAC and Cipher feedbacks are indelibly recorded in the entropy of the Register Bank, constantly diffusing at each Sample clock. The abundance of tainted variables precludes the chance of meaningful collision.

The Register Bank- the Source

As seen in Figure 4, the Register Bank consists of 4 tiers, each of which generates difficult to analyze pseudo-random words.

The statistics of the Top, Middle and Bottom tiers input to the MAJ combiner are degraded by feedback. With random slip complemented feedbacks, the sequence is non-linear and cannot be predicted accurately. We know that the MAJ output is typically biased, where the bias is considerably lowered by XORing its output to its 7 right rotated image.

The MAJ combiner XORed to its image is further XORed to the output of the Super Tier. The Super Tier is always activated, and is always XORed to its 7 left rotated image, so that the combination is typically well balanced (only slightly biased).

This last combination from the 4 tier Hybrid Combiner is a well balanced palliative that rains down to the cipher mask, improving the statistics on XOR vectors at every step, hiding the biasing effect of feedback and non-linear permutations.

Concluding the Overview of Data Manipulation

We have shown the massive random diffusion of external encoded data variables into other variables, wherever they are and from wherever a changing input is derived. The 128 bit Register Bank is the ultimate store of unpredictability, supported, and protected by the Data Churn whose Cipher Mask encodes and digests each message word back into the Register Bank "for future reference" and back into the Data Churn and Feedback Processor for future secured encoding. We have not delved into the Random Controller interaction with the 32 bit Data Word Manipulator. The effects of the 17 separate permutations on the result word are the basis for the single step (clock) highest mark [7] result Cipher Mask; producing a new unpredictable 32 bit word at every step, without the benefits of feedback.

MAC feedback is especially useful (and important), not only for Data Authentication, but also for initializing stream ciphers with secret keys and initial values, and is also important in instances where we may choose to accelerate initialization of random seeds for "delayed deterministic" random number generation.

We have not included a discussion of the ZK-Crypt random Frequency Modulated noise source for TRNGs, despite NIST's request to combine Random Number generation with Data Authentication. The present design followed the German AIS 31 recommendation for on-line provable noise sources for random number generation. The AIS 31 spec demands proof that as the noise source drives the deterministic random number generator it loads the generator with redundant entropy; and, in parallel, the CPU ascertains proof of a minimal threshold of randomness [3]. The Control Units with permutation Encoders, driving the Register Bank and the Data Churn are the well proven Deterministic Random Number Generator which is the basis for the ZK-Crypt candidacy in the EU eSTREAM contest and is compatible with the German AIS 20 certifying DRNG specification.

Current and Energy Consumption

The ZK-Crypt is a compact module with a high transistor toggle rate. Consequently, when used as a slow running Data Authenticator, its current consumption is essentially the integral of a series spikes over a normalized interval. In all but a few exceptional instances the height of the spike can be reduced with a commensurate lowering of its maximum value of, and with a slight broadening of the spike, to match the natural capacitance of the device.

The average energy consumption is linearly affected by the Sampling rate of the host. Total energy consumption per encoded bit is close to constant so that in sensitive applications; e.g., contactless smart cards, users may have to limit the sampling frequency to the integral of the spike area over an amenable period.

The ASIC design department of the ETH in Zurich integrated and compared the eight most promising eSTREAM stream cipher hardware designs on a trial wafer. The ZK-Crypt version 1 ranked highest by a very large margin, in throughput and lowest in energy consumption per encoded bit for 128 bit key lengths devices (the other two leading devices had 80 bit keys) [8].

Initializing the MAC

Initializing the ZK-Crypt for any deterministic or random process is easily accomplished. We will show in the next section, in detail, two robust methods using three of the five programmable command configurations.

1) The global Reset command initializes all variables; obscure and directly loadable. (Many flip-flops are purposely initialized at Reset to logic "1".) This single command may suffice for initializing un-keyed hashing. See Hash Initialization in Figure 9.
2) After the global Reset Command, 128 variable bits, including three 32 celled tiers (Top, Middle and Bottom) in the Register Bank and 32 of the variables in the Random Controller are directly Host loadable with four load enabling commands. (See first step loading in Keyed Hash Initialization in Figure 9.)
3) MAC mode feedback implicit key randomization of all 334 MAC mode variables (including obscure variables, the loadable Registers and control variables mentioned above and the Result buffer) may be implemented by inserting any defined number of key words in the Data In "Message" port, each word inserted at least one half clock cycle prior to a Sample command. Implicit randomization must follow a Reset command for deterministic loading. This is arguably the simplest, and possibly the best single method for key loading. Cryptanalysts prefer "hybridic" initializing, where the first step is loading tiers and control variables, as in paragraph 2), followed by Implicit key randomization outlined in this section. (Equivalent to Data Digest Encoding shown in Figure 9.)
4) Multi-step Cipher Feedback pseudo-randomization after optional key loading and first step Reset is accomplished by exercising the Sample command a defined number of times in the Cipher Feedback configuration. (Following Loading sequence of Keyed Hash Initialization and also in the Interhash Cipher Feedback Digest in Figure 9.)
5) Lower power multi-step without feedback pseudo-randomization after optional key loading and first step Reset is accomplished by exercising the Sample command a defined number of times without feedback. (Not shown in Figure 9.)

Keyed MAC and Un-Keyed Hashing Procedures

Figure 9 depicts the proposed ZK-Crypt Hash/MAC flow for Hash and MAC data authentication procedures. Each block depicts the ZK-Crypt Engine following a command, e.g., E_{L3} is the condition of the engine after loading the Bottom tier key word (the last load command).

The Keyed Hash Initialization procedure consists of global Resetting of the Engine, Loading 128 bits of key, and performing 8 Rounds of digesting using Sparse Cipher Feedback.

Un-keyed Hash Initialization consists simply of a global Resetting of the Engine, leaving a resident all "1" output from the Feedback Store.

MAC and Hash Digest encoding are identical. Header, "Message" and Tail Words are encoded by the Cipher Mask and recycled in MAC Feedback mode.

Following the MAC/Hash Digest, 16 rounds of randomizing using Cipher Feedback is recommended. At stage EIH15, the Engine has digested the Header, the Data File and the Tail; and completely randomized variables, ready for Tagging. The ZK-Crypt engine at stage E_{MT0} is ready for any practical size tag.

In the MAC/Hash TAG sequence tag words H_0 to H_t are generated using Sparse Cipher Feedback.

The ZK-Crypt - Performance & Size

Size- For a full hardware implementation which includes a Finite State Machine, operative to enable high speed DMA and pipelining, True Random Number Generation, Stream Ciphering and Data Authentication activation– the ZK-Crypt uses less than 8000 2inNAND equivalent gates, occupying less than 1/7 mm^2 with the present popular 0.18 µ technology.

Pipe Lined Throughput at Standard Single Step Operations- For Data Authentication, Stream Ciphering, and True Random Number Generation, generating one 32 bit word at every clock (machine cycle) the ZK-Crypt typically will output 3 GBits/Sec at 100 MHz.

Comparing ZK-Crypt to AES (in Hardware)

The ZK-Crypt is many times more compact with less than one eighth of the gate count and is significantly faster than all versions of the AES in hardware; thereby anticipating that when the ZK-Crypt goes on line in 2007 with 0.09µ technology, at 1.8 volts- it will benchmark at 1,150 [MBit/mWatt Sec] against AES (without DPA countermeasures) with 13 [MBit/mWatt Sec] (or less than 4 [MBit/mWatt Sec] with DPA countermeasures) or A/51 with about 273 [MBit/mWatt Sec].

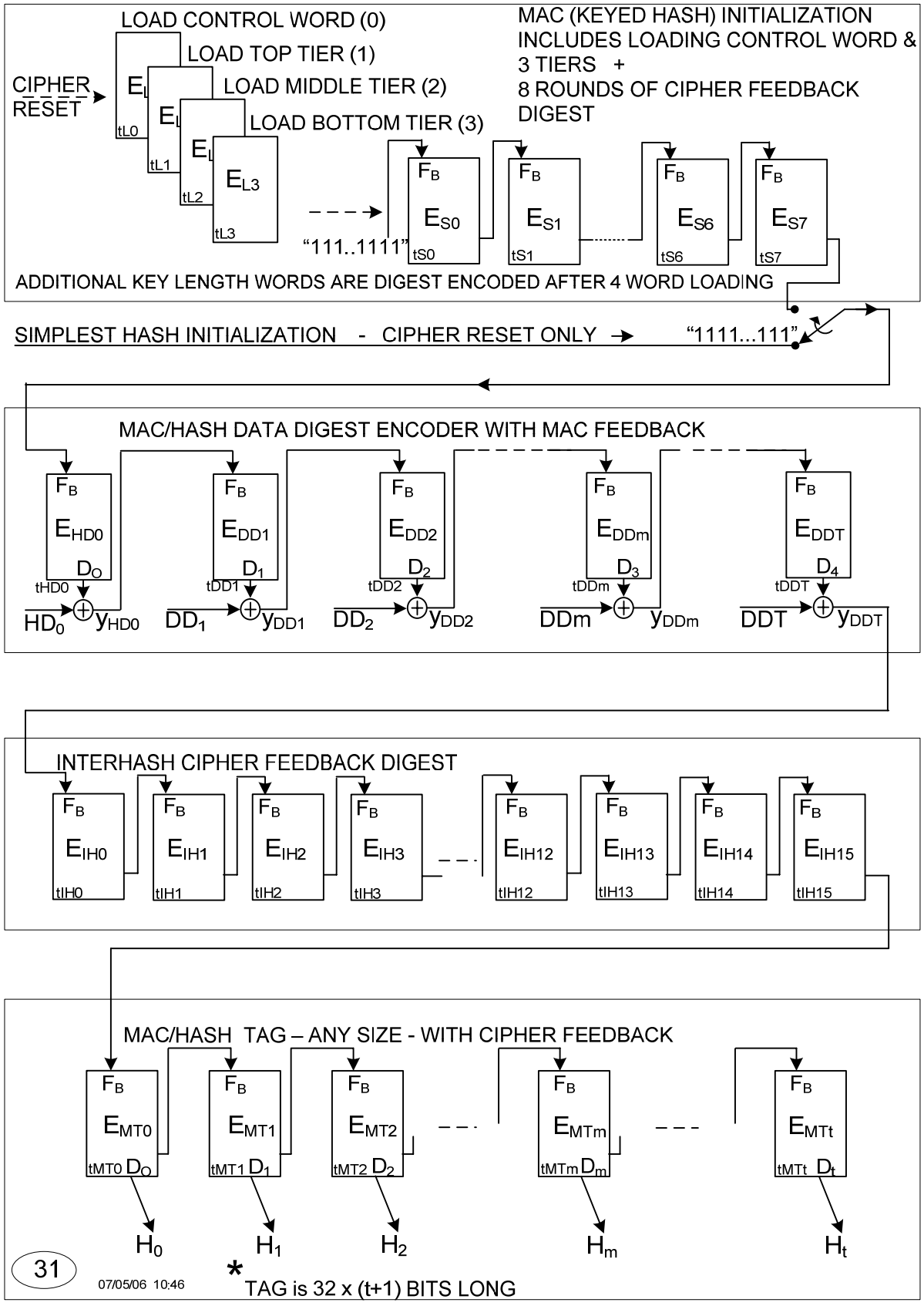

Figure 9 - Keyed and Un-keyed Data Authentication Sequence [1]

Bibliography

[1] ZK-Crypt Circuit & Concept Drawings vers 3, FortressGB Prespec, FortressGB.com, May 2006.
[2] Comparing the ZK-Crypt vers. 2 to the NIST-AES Cipher, FortressGB.com, February 2006.
[3] The Dual Clock Noise Source, FortressGB Prespec, FortressGB.com, March, 2006.
[4] O. Dunkelman & A. Hecht, Security Analysis – ZK-Crypt, vers 2, FortressGB.com, February 2006.
[5] O. Dunkelman & A. Hecht, The ZK-Crypt Algorithm & Implementation vers. 2, FortressGB.com, Feb. 2006.
[6] The Host sees the ZK-Crypt vers. 3, FortressGB.com, May 2006.
[7] George Marsaglia DieHard test suites are available at ftp://ftp.csis.hku.hk/pub/random/source.
[8] SASC 2006-Stream Ciphers Revisited, Workshop Record, February 2006, ESAT, Leuven, page 123.

The ZK-Crypt is the subject of two PCT pending patent applications.

The Authors:

Carmi Gressel is presently the CTO of FortressGB Ltd, London. He was the chief architect of hardware modular exponentiators and firmware protocols which Fortress U&T Ltd. and M-Systems sold to ST Microelectronics, Motorola Semiconductor, NEC Electronics, Samsung Semiconductor and others. He has been active for the last 18 years designing security hardware and was the founder of Fortress U&T, now a wholly owned subsidiary of M-Systems. Carmi is the author of 12 issued patents in security designs, protocols and algorithmic implementations.

Avi Hecht is presently FortressGB Ltd.'s field application engineer, principally working on hardware and firmware security systems for sports stadiums, based on contactless smart cards and other portable devices. Avi has been very active in designing the enhancements in ZK-Crypt versions 2 and 3. Avi is a co-author of 3 FortressGB patent applications.

Orr Dunkelman is a cryptanalyst presently a lecturer on cryptography in the Computer Science Department of the Technion. Orr did the first thorough cryptanalysis of the ZK-Crypt, and worked with Carmi and Avi on complexity enhancements of versions 2 and 3. Orr is a reviewer for 5 refereed journals in the field of cryptography, information theory and security, and has been on the program committee of numerous cryptographic conferences. Orr has done cryptanalysis and developed new methods of attack on Serpent, SC2000, SHACAL-1, COCONUT98, AES and Kasumi and is coauthor of two ZK-Crypt patent applications.

Ran Granot, a manager at FortressGB, and previously the CEO of Fortress U&T. Ran was actively involved in the development of security systems and hardware solutions. Ran is the project manager of ZK-Crypt concepts and solutions, and directs the system architecture group of FortressGB.